

combine

```

$TITLE COMBrttp
*
*           GAMS RESOURCE ALLOCATION / SCHEDULING MODEL  (COMBINE)
*
* This version combines the flow and schedule models into one integrated model
* that accounts for the availability of personnel while optimizing flow
* -it also uses individual rates, and has constraints to ensure a minimum amount
*   is spent in a task before someone can change tasks
$INLINECOM { }
  SETS
    P      People
  /
  $include'peoplein.prn';
  /
    PT     Person Type
  /
  $include'ptype.prn';
  /
    TK     Task
  /
  NO_WORK
  $INCLUDE'tasks.prn';
  /
    CHECKTK(TK)
  /
  $INCLUDE'checktk.prn';
  /
    TP     Time Period
  /
  $INCLUDE'time15.prn';
  /
    S      Shift Identifier
  /
  $include'sdef.prn';
  /
    ALIAS(TK, KK)
    ALIAS(TP, STRT, FIN, TPP);

SETS  PDATA(P,PT,S)   Person to Type to Shift mapping
  /
  $include'pdata.prn';
  /
    SDATA(S,STRT,FIN) Shift Data
  /
  $include'sdata.prn';
  /
  PPTMAP(P,PT)  person to person type map
  PTPMAP(P,TP)  person to time period availability map
  CHKPTP(P,TP)  defined if someone can check during time period TP
  WP(S,TP,TPP)  defines min TPs that P must work in TK if starts TK in TP
  WP1(P,TP,TPP) defines TPP's that person can work
  PSMAP(P,S)    person to shift map

```

combine

TABLE TKMAP(TK,TK) percentage flows from task to task  
\$INCLUDE 'flow.prn';

SCALAR

TPHR time periods per hour /4/  
CHKHR minimum checking hours per pharmacist /2/

PARAMETERS

RATEHR(P,TK) Rate (RXs) per person per hour for each Task TK  
DRATEHR(TK) Rate (RXs) per person per hour for each Task TK  
LABORHR(PT) Cost per person of type PT per hour  
LABOR(P,TK) Cost per person P per time period  
RATE(P,TK) (RXs) per person per time period for each Task TK  
RX0(TK,TP) Initial # of Rx's in each Task (INPUT TO MODEL)  
MAXPTP(TP) maximum total # of people allowed during time period TP  
MAXPTK(TK,TP) maximum people allowed in Task during any time period  
RXCOST(TK) RX cost per unit time in task TK  
AVAIL(P,TP,TK) availability of person P in Task TK at time TP  
TPPTP(TP) total people in time period (AFTER SOLUTION FOUND)  
TPAVAIL(TK,TP) total people available in task TK during time period TP  
PRFCST(P,TK) task preference cost per person -- specified  
TIMES(TK) earliest time that someone can be scheduled to task TK  
TIMEE(TK) latest time that someone can be assigned to task TK  
NAV(TK) indicates whether task TK can be assigned at all  
CHKELG(P) indicates if a pharmacist is eligible to do a checking function  
PAVAIL(P) indicates if person P is available at any time during the day  
MINTP minimum TPs before a person can change tasks  
COUNT used as a counter  
COUNT1  
BREAK(TP,S) specifies break and lunch times for shift S  
maxtp  
TPEOP(TK,TP) total people assigned to TK during TP in solution  
TPCHK(P) number of timeperiods that a person is available to do checkin  
g  
MINTPCHK(P) minimum number of time periods a person must check

\*\*\* The following figures out TP availability of shifts -----\*\*

SETS

STRTTP(TP,S) starting time period for shift S  
FINTP(TP,S) finishing time period for shift S  
STPMAP(TP,S) shift to time period map

PARAMETERS

START(S) order(sequence) of starting time period for shift S  
FINISH(S) order(sequence) of finishing time period for shift S

BREAK(TP,S)=0;

\$INCLUDE'SPARAMS.PRN'; {file has lunch and break information for each shift}

STRTTP(TP,S) = SUM(FIN, SDATA(S,TP,FIN) );

FINTP(TP,S) = SUM(STRT, SDATA(S,STRT,TP) );

combine

```

LOOP((TP,S)$STRTTP(TP,S),
  START(S) = ORD(TP);
);
LOOP((TP,S)$FINTP(TP,S),
  FINISH(S) = ORD(TP);
);
STPMAP(TP,S)$ (ORD(TP) GE START(S) AND ORD(TP) LE FINISH(S)
  AND NOT BREAK(TP,S)) = 1;
STPMAP(TP,S)$ (START(S) EQ FINISH(S)) = 0;
***-----***

PTPMAP(P,TP) = SUM((PT,S)$PDATA(P,PT,S),STPMAP(TP,S)) ;
PSMAP(P,S) = SUM(PT$PDATA(P,PT,S), PDATA(P,PT,S));
PRFCST(P,'NO WORK')=1;
$INCLUDE'PREF.PRN';      {file describes preference list for each person}
AVAIL(P,TP,TK)$PRFCST(P,TK) = PTPMAP(P,TP);

TIMES(TK)=1;              {  THESE ARE INITIALIZATIONS STATEMENTS  }
TIMEE(TK)=CARD(TP);       {                                AND MUST BE BEFORE                                }
NAV(TK)=0;                {                                }
MAXTP = 96;               {                                PARAMS.PRN                                }
MINTP = 1;               {                                IS READ                                }
MAXPTK(TK,TP)=900;        {                                }
$INCLUDE 'PARAMS.PRN';    { also includes times and timee information}
AVAIL(P,TP,TK)$ (NAV(TK) OR ORD(TP) LT TIMES(TK) OR ORD(TP) GT TIMEE(TK)) = 0;
MAXPTK(TK,TP)$ (NAV(TK) OR ORD(TP) LT TIMES(TK) OR ORD(TP) GT TIMEE(TK)) = 0;
TPAVAIL(TK,TP) = SUM(P,AVAIL(P,TP,TK));
MAXPTK(TK,TP)=MIN(MAXPTK(TK,TP),TPAVAIL(TK,TP));

*-----THE FOLLOWING FIGURES OUT TP GROUPINGS FOR MINIMUM TIME IN TASK-----*
LOOP(S,
  COUNT = 0;
  LOOP(TP$STPMAP(TP,S),
    IF(MOD(COUNT,MINTP)=0,
      COUNT1 = 0;
      LOOP(TPP$(STPMAP(TPP,S) AND ORD(TPP) GT ORD(TP)),
        COUNT1 = COUNT1 + 1;
        IF((COUNT1 LT MINTP), WP(S,TP,TPP) = YES );
      );
      IF((COUNT1 LT (2*MINTP-1)),
        WP(S,TP,TPP)$ (STPMAP(TPP,S) AND ORD(TPP) GT ORD(TP))=YES;
      );
    );
  COUNT = COUNT + 1;
);
WP1(P,TP,TPP) = SUM(S$PSMAP(P,S), WP(S,TP,TPP));
display wp,wp1;
*-----*

*-----THE FOLLOWING COULD BE SPECIFIED IN THE INTERFACE/EXTERNAL FILE-----*
LABORHR(PT)      = 11;              {initialize labor cost}
LABORHR('Clerical') = 7.5;

```

combine

```
LABORHR('Technician') = 11;  
LABORHR('Pharmacist') = 27;
```

\*-----\*

```
PPTMAP(P,PT) = SUM(S$PDATA(P,PT,S), PDATA(P,PT,S) );  
PAVAIL(P) = SUM((TP,TK)$AVAIL(P,TP,TK),AVAIL(P,TP,TK));  
LABOR(P,TK) = SUM(PT$PPTMAP(P,PT), LABORHR(PT)) / TPHR;  
LABOR(P,'NO_WORK') = LABOR(P,'NO_WORK') - (.1*(1-.01*CARD(TK)) );
```

```
$INCLUDE'PRATE.PRN';  
RATEHR(P,TK)$ (PRFCST(P,TK) AND RATEHR(P,TK) = 0) = DRATEHR(TK);  
  RXCOST(TK) = 1-.01*ORD(TK);  
  MAXPTP(TP) = 900;  
  RATE(P,TK) = RATEHR(P,TK) / TPHR;
```

```
CHKELG(P)$PPTMAP(P,'Pharmacist') = SUM(CHECKTK(TK),PRFCST(P,CHECKTK));  
CHKPTP(P,TP)$CHKELG(P) = SUM(CHECKTK,AVAIL(P,TP,CHECKTK));  
TPCHK(P)$CHKELG(P) = SUM(TP$CHKPTP(P,TP),CHKPTP(P,TP));  
*DISPLAY TPCHK;
```

```
MINTPCHK(P) = MIN(TPCHK(P),TPHR * CHKHR);
```

#### POSITIVE VARIABLES

```
  RX(TK,TP) # of Rx's in task TK during time period TP  
  RXPROC(TK,TP) # of Rx's PROCESSED in task TK during time period TP  
  SCHED(P,TP,TK) equals 1 if person P is assigned to task TK during time perio  
d TP  
;
```

#### \*INTEGER VARIABLES

```
*;
```

#### VARIABLE

```
  OBJ;
```

#### EQUATIONS

```
  COST                objective function  
  RXPRC(TK,TP)        calculates # of Rxs processed during time period TP  
  RXPRC1(TK,TP)       added condition to calculate # of Rxs processed during TP  
  RXQ(TK,TP)          calculates total # of Rxs in task TK during time period TP  
  MXPTK(TK,TP)        maximum # of people in task  
  SUPPLY(P,TP)        each person can only be assigned 1 task at a given time  
  TEAMPC(TK,TP)       team functions must have the same # of pharmacists as clericals  
  CHKMIN(P)          each pharmacist must check at least 2 hrs a day  
  TKMIN(P,TP,TPP,TK) forces person in task to stay for min TPs before changing  
  TKMAX(P,TK)         maximum number of time periods a person can do a task  
;
```

```
COST..  OBJ =E= SUM((P,TP,TK)$AVAIL(P,TP,TK),  
  (LABOR(P,TK) + PRFCST(P,TK)) * SCHED(P,TP,TK) )  
  + SUM((TK,TP)$ (ORD(TK)<CARD(TK)),RXCOST(TK)*RX(TK,TP));
```

```
*sched.up(p,tp,tk)$ (NOT AVAIL(P,TP,TK)) = 0;
```

```

                                combine
*RXPROC.FX(TK,TP)$ (MAXPTK(TK,TP)= 0) = 0;
RXPRC(TK,TP)..    RXPROC(TK,TP) =L= SUM(P$AVAIL(P,TP,TK),
                                SCHED(P,TP,TK) * RATE(P,TK));
RXPRC1(TK,TP)..   RXPROC(TK,TP) =L= RX(TK,TP);

RXQ(TK,TP)..      RX(TK,TP) =E= RX(TK,TP-1) + RX0(TK,TP) +
                                SUM(KK$TKMAP(KK,TK), TKMAP(KK,TK)*RXPROC(KK,TP-1) )
                                - SUM(KK$TKMAP(TK,KK), TKMAP(TK,KK)*RXPROC(TK,TP-1) );
MXPTK(TK,TP)..    SUM(P$AVAIL(P,TP,TK), SCHED(P,TP,TK) ) =L= MAXPTK(TK,TP);
SUPPLY(P,TP)$PTPMAP(P,TP).. SUM(TK$AVAIL(P,TP,TK), SCHED(P,TP,TK) ) =E= 1;

```

\*--- the following constraints are "hard coded" into the program -(5/6/97)-\*\*

```

TEAMPC(TK,TP)$ (CHECKTK(TK) AND MAXPTK(TK,TP) GT 0)..
                                SUM(P$PPTMAP(P,'Pharmacist'),SCHED(P,TP,TK)) -
                                SUM(P$PPTMAP(P,'Clerical'),SCHED(P,TP,TK)) =E= 0;

```

```

CHKMIN(P)$ (CHKELG(P) and PAVAIL(P))..
                                SUM((TP,CHECKTK)$ (PRFCST(P,CHECKTK) and AVAIL(P,TP,CHECKTK)),
                                SCHED(P,TP,CHECKTK)) =G= MINTPCHK(P);

```

\*--- the following constraints ensure that people stay in task for minimum time

```

TKMIN(P,TP,TPP,TK)$ (PAVAIL(P) AND WP1(P,TP,TPP)
                                AND PRFCST(P,TK) AND AVAIL(P,TP,TK) AND AVAIL(P,TPP,TK))..
                                SCHED(P,TP,TK) - SCHED(P,TPP,TK) =E= 0;

```

```

TKMAX(P,TK)$ (PAVAIL(P) AND PRFCST(P,TK))..
                                SUM(TP$AVAIL(P,TP,TK), SCHED(P,TP,TK)) =L= MAXTP;

```

```

$OFFSYMXXREF OFFSYMLIST
MODEL PSCHED /ALL/;
OPTION ITERLIM = 950000;
OPTION RESLIM = 54000;
OPTION LIMROW =3;
OPTION LP = CPLEX;
OPTION optca = 100;
*PSCHED.optfile =1;
SOLVE PSCHED USING LP MINIMIZING OBJ;

```

```

TPEOP(TK,TP)=SUM(P, SCHED.L(P,TP,TK) );

```

```

FILE REP1 /PHARM3A1.TXT/;
FILE REP2 /PEOPLE.prn/;
FILE REP3 /RXPROC.prn/;
FILE REP4 /RXCUM.prn/;

```

```

PUT REP1;
REP1.PW=500;
PUT "OBJECTIVE VALUE":18, OBJ.L:10:2 /;
PUT "ABSOLUTE GAP    ":18,PSCHED.OPTCA:15:3 /;
PUT "RELATIVE GAP    ":18,PSCHED.OPTCR:8:3 //;

```

combine

```

PUT "TIME      ","      ","      "      RX IN  ","      RX'S      ","      "/;
PUT "PERIOD    ","TASK  ","      "      QUEUE  ","PROCESSED","      PEOPLE"/;
LOOP( (TP,TK)$RX.L(TK,TP),
      PUT TP.TL:8, TK.TL:8, RX.L(TK,TP):8:1,RXPROC.L(TK,TP):10:1,
      TPEOP(TK,TP):8:1  /;
      );
PUT //;

```

```
PUT "TIME      ", "TOTAL " /;
PUT "PERIOD    ", "PEOPLE" /;
```

```

LOOP(TP,
      TPPTP(TP) = SUM(TK, TPEOP(TK,TP) )
      PUT TP.TL:8, TPPTP(TP):8 /;
    );
PUT /;

```

```

PUT REP2;                                { PERSONNEL ALLOCATION REPORT }
REP2.PW=255;
*REP2.PC = 6;

PUT "*" PERSONNEL ALLOCATION PER TIME PERIOD" /;
PUT " ";
LOOP(TK$(ord(TK)<18),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
    LOOP(TK$(ord(TK)<18),
        PUT ROUND(TPEOP(TK,TP)+0.499):11:1;
    );
    PUT /;
);
PUT " + " /;
PUT " ";
LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
    LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
        PUT ROUND(TPEOP(TK,TP)+0.499):11:1;
    );
    PUT /;
);
PUT /;

```

combine

```
PUT REP3;
REP3.PW=500;
PUT "  # OF RX'S PROCESSED IN EACH TASK DURING EACH TIME PERIOD" /;
PUT "          ";
LOOP(TK$(ord(TK)<18),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
    LOOP(TK$(ord(TK)<18),
        PUT RXPROC.L(TK,TP):11:1;
    );
PUT /;
);
PUT /;
PUT "          ";
LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
    LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
        PUT RXPROC.L(TK,TP):11:1;
    );
PUT /;
);
PUT /;

PUT REP4;
REP4.PW=500;
PUT "TOTAL # OF RX'S IN QUEUE FOR EACH TASK DURING EACH TIME PERIOD" /;
PUT "          ";
LOOP(TK$(ord(TK)<18),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
    LOOP(TK$(ord(TK)<18),
        PUT RX.L(TK,TP):11:1;
    );
PUT /;
);
PUT /;
PUT /;
PUT "          ";
LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
    PUT TK.TL:11;
);
PUT /;
LOOP(TP,
    PUT TP.TL:4;
```

```

                                combine
LOOP(TK$(ord(TK)>=18 AND ORD(TK)<CARD(TK)),
  PUT RX.L(TK,TP):11:1;
);
PUT /;
);
PUT /;

```

```

$INCLUDE'PSCHEDFT.RPG'

```

```

* file testrep /test.out/;
  file DUMMY /!PSCHED/;
* put testrep;
* loop((p,tk,tp)$sched.1(p,tp,tk),
*   put p.tl:5, tp.tl:6, tk.tl:12, sched.1(p,tp,tk) /;
* );

```

```

put DUMMY;
PUT PSCHED.MODELSTAT /;
PUT$(PSCHED.MODELSTAT EQ 1) "OPTIMAL"/;
PUT$(PSCHED.MODELSTAT EQ 2) "LOCALLY OPTIMAL"/;
PUT$(PSCHED.MODELSTAT EQ 3) "UNBOUNDED"/;
PUT$(PSCHED.MODELSTAT EQ 4) "INFEASIBLE"/;
PUT$(PSCHED.MODELSTAT EQ 5) "LOCALLY INFEASIBLE"/;
PUT$(PSCHED.MODELSTAT EQ 6) "INTERMEDIATE INFEASIBLE"/;
PUT$(PSCHED.MODELSTAT EQ 7) "INTERMEDIATE NONOPTIMAL"/;
PUT$(PSCHED.MODELSTAT EQ 8) "INTEGER SOLUTION"/;
PUT$(PSCHED.MODELSTAT EQ 9) "INTERMEDIATE NON-INTEGERS"/;
PUT$(PSCHED.MODELSTAT EQ 10) "INTEGER INFEASIBLE"/;
PUT$(PSCHED.MODELSTAT EQ 12) "ERROR UNKNOWN"/;
PUT$(PSCHED.MODELSTAT EQ 13) "ERROR NO SOLUTION"/;

```



\* REPORT GENERATOR FOR GANNT CHART (IN GAMS)  
FILE REP99 /ASSIGN1.PRN/;

PARAMETER COUNT;

```
PUT REP99;
REP99.PW = 500;
PUT "OBJECTIVE VALUE":18, OBJ.L:10:2 /;
PUT "ABSOLUTE GAP    ":18, PSCHED.OPTCA:15:3 /;
PUT "RELATIVE GAP    ":18, PSCHED.OPTCR:8:3 //;

PUT "  PERSONNEL ALLOCATION PER TIME PERIOD" /;
PUT "                                ";
LOOP (TP,
  IF (ORD(TP) = 44, PUT / );
  PUT TP.TL:5
);
PUT /;
```

```
LOOP ((P,PT)$PPTMAP(P,PT),
  PUT P.TL:11,P.TE(P):19," ",PT.TL:11;
  LOOP (TP,
    COUNT = 0;
    IF (MOD(ORD(TP),19) = 0, PUT / );
    IF (NOT PTPMAP(P,TP), PUT "N/A ":11;
    ELSE
      LOOP (TK$SCHED.L(P,TP,TK),
        COUNT = COUNT + 1;
        IF (COUNT = 1,
          PUT TK.TL:11;
        );
      );
    );
  );
  PUT /;
);
```